AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Previously Presented) A system for designing a configurable processor, the system comprising:

hardware generation means for, based on a configuration specification including a predetermined portion and a user-defined portion, generating a description of a hardware implementation of the processor, the predetermined portion specifying a configuration of a core register file, and the user-defined portion specifying whether to include a user-defined register file in the processor in addition to the core register file; and

software generation means for, based on the configuration specification, generating software development tools specific to the hardware implementation;

wherein the hardware generation means includes register generation means for, based on the user-defined portion of the configuration specification, generating a description of the user-defined register file separate from and in addition to a description of the core register file in the description of the hardware implementation of the processor; and

the software generation means is for, based on the user-defined portion, including software related to the user-defined processor register file in the software development tools.

2. (Original) The system of claim 1, wherein the software related to the user-defined processor register file includes an instruction for accessing elements in the register file according to a field of the instruction.

3. (Original) The system of claim 2, wherein the hardware generation means is for generating at least part of the description of the hardware implementation in a register transfer level hardware description language.

4. (Previously Presented) The system of claim 1, wherein the configuration specification defines the user-defined register file using a statement specifying the width of elements in the user-defined register file.

5. (Previously Presented) The system of claim 1, wherein the configuration specification defines the user-defined register file using a statement specifying the number of elements in the user-defined register file.

6. (Previously Presented) The system of claim 1, wherein the hardware generation means is for determining a number of at least one of read ports and write ports of the user-defined register file independently of the configuration specification.

7. (Canceled)

8. (Original) The system of claim 1, wherein the hardware generation means is for generating, as part of the processor hardware implementation description, a description of logic to assign write ports of the user-defined register file to instruction operands to minimize data staging costs.

9. (Previously Presented) The system of claim 1, wherein the hardware generation means is for generating pipeline logic for accessing the user-defined register file.

10. (Original) The system of claim 9, wherein read ports for the user-defined register file are read in the earliest stage of any instruction that uses them as a source operand.

11. (Original) The system of claim 9, wherein write ports for the user-defined register file are read in the latest stage of any instruction that uses it as a destination operand or in an instruction commit stage if later.

12. (Previously Presented) The system of claim 1, wherein the hardware generation means is for generating, as part of the hardware implementation of the processor, logic to provide a read

port for the user-defined register file for each field, within an instruction accessing the user-defined register file, used to select a source operand from the user-defined register file.

13. (Previously Presented) The system of claim 1, wherein the hardware generation means is for generating, as part of the hardware implementation of the processor, bypass logic for accessing the user-defined register file.

14. (Original) The system of claim 13, wherein the hardware generation means is for generating the interlock logic for a given pipeline of the processor described by the configuration specification based on instruction operand and state usage descriptions in the configuration specification.

15. (Previously Presented) The system of claim 1, wherein the hardware generation means is for generating, as part of the hardware implementation of the processor, interlock logic for accessing the register file.

16. (Original) The system of claim 15, wherein the hardware generation means is for generating the interlock logic based on scheduling information in the configuration specification.

17. (Original) The system of claim 15, wherein the hardware generation means is for generating the interlock logic for a given pipeline of the processor described by the configuration specification based on instruction operand and state usage descriptions in the configuration specification.

18. (Original) The system of claim 1, wherein the hardware generation means is for generating the processor hardware implementation description to use at least one portion of processor logic described by the predetermined portion of the configuration specification to support access of the user-defined register file.

19. (Original) The system of claim 18, wherein the at least one portion of processor logic includes address computation logic.

20. (Original) The system of claim 19, wherein the address computation logic includes address adder logic.

21. (Previously Presented) The system of claim 19 wherein the at least one portion of processor logic includes data alignment logic shared between the portions of the processor corresponding to predetermined and user-defined portions of the configuration specification.

22. (Original) The system of claim 19 wherein the at least one portion of processor logic is a data memory.

23. (Original) The system of claim 1, wherein the user-defined portion of the configuration specification includes a description of an instruction which conditionally writes to the user-defined register file.

24. (Original) The system of claim 1, wherein the software generation means is for generating, as part of the software relating to the user-defined register file, diagnostic tests for design verification and manufacturing of the processor based on the configuration specification.

25. (Original) The system of claim 1, wherein:

the configuration specification includes both reference and implementation semantics for instructions of the processor; and

the reference semantics can be used to verify design correctness of the implementation semantics.

26. (Original) The system of claim 1, wherein:

the processor instruction set description language includes instruction test cases; and

the software generation means is for generating diagnostics for the test cases.

27. (Original) The system of claim 1, wherein the software generation means is for automatically generating test vectors by sampling operands to instructions in the processor instruction set description language while running an application.

28. (Original) The system of claim 1, wherein the software generation means is for generating at least a portion of an operating system as part of the software relating to user-defined states and register files.

29. (Original) The system of claim 28, wherein the generated portion of the operating system includes save and restore sequences for processor state.

30. (Original) The system of claim 29, wherein the save and restore sequences are generated with respect to interdependencies of component states and is valid for those interdependencies.

31. (Previously Presented) The system of claim 28, wherein the operating system is capable of saving less than an entirety of processor state in accordance with a dynamic reference by a new task after switching contexts.

32. (Original) The system of claim 28, wherein:

the user-defined portion of the configuration specification defines a software data type not found in the predetermined portion of the configuration specification; and

the compiler supports the software data type.

33. (Original) The system of claim 1, wherein the software generation means is for generating at least one of a compiler, a linker, a simulator and a debugger as part of the software relating to the user-defined register file.

34. (Previously Presented) The system of claim 1, wherein:

the software generation means is for generating a compiler as part of the software relating to the user-defined register file; and

the compiler is capable of allocating program variables to registers in the user-defined register file.

35. (Original) The system of claim 34, wherein the compiler is further capable of loading a value from memory into a register in the user-defined register file, and storing a value in a register of the user-defined register file into memory.

36. (Original) The system of claim 34, wherein the compiler is further capable of moving a value from one register in a user-defined register file to another register in a user-defined register file.

37. (Original) The system of claim 34, wherein the compiler is for using scheduling information in the configuration specification to determine stall cycles of instructions in the software generated by the software generation means which access the user-defined register file.

38. (Original) The system of claim 1, wherein the software generation means is for automatically generating a monitor to check for coverage of bypass paths.

39. (Previously Presented) A system for designing a configurable processor, the system comprising:

hardware generation means for, based on a configuration specification including a predetermined portion and a user-defined portion, generating a description of a hardware implementation of the processor; and

software generation means for, based on the configuration specification, generating software development tools specific to the hardware implementation;

wherein the configuration specification includes a statement specifying scheduling information of instructions used in the software development tools;

the hardware generation means is for, based on the statement in the configuration specification, determining whether and how to generate a description of at least one of pipeline logic, pipeline stalling logic and instruction rescheduling logic.

40. (Original) The system of claim 39, wherein the scheduling information includes a statement that an operand of an instruction enters a pipeline of the processor at a given stage.

41. (Original) The system of claim 39, wherein the scheduling information includes a statement that an operation of an instruction exits a pipeline of the processor at a given stage.

42. (Original) The system of claim 39, wherein:

the software generated by the software generation means includes a compiler which uses instructions described in the user-defined portion of the configuration specification; and

the compiler uses the scheduling information during instruction scheduling to schedule the instructions described in the user-defined portion of the configuration specification.

43. (Original) The system of claim 39, wherein the configuration specification includes a description of an instruction which requires a plurality of processor cycles to be processed.

44. (Original) The system of claim 43, wherein:

the configuration specification includes a description of an instruction's semantics which is independent of a target pipeline of the processor; and

the hardware generation means is for generating as part of the processor hardware implementation a pipeline based on a pipeline description separate from the instruction semantics.

45. (Original) A system for designing a configurable processor, the system comprising:

hardware generation means for, based on a configuration specification including a predetermined portion and a user-defined portion, generating a description of a hardware implementation of the processor;

software generation means for, based on the configuration specification, generating software development tools specific to the hardware implementation; and

document generation means for generating documentation of a processor instruction set described by the configuration specification based on the configuration specification.

46. (Original) The system of claim 45, wherein the document generation means is for using reference semantics of instructions defined in the configuration specification to generate the processor instruction set documentation.

47. (Original) The system of claim 45, wherein:

the user-defined portion of the configuration specification contains reference semantics of an instruction defined therein and a user-defined specification of at least one of a synopsis and a text description for the user-defined instruction; and

the document generation means is for using the at least one of the synopsis and the text description to generate documentation of the processor instruction set.

48. (Previously Presented) A system for designing a configurable processor, the system comprising:

hardware generation means for, based on a configuration specification including a predetermined portion and a user-defined portion, generating a description of a hardware implementation of the processor; and

software generation means for, based on the configuration specification, generating software development tools specific to the hardware implementation;

wherein the user-defined portion of the configuration specification includes a user-defined specification of a processor exception and when a processor instruction raises the exception; and

the hardware generation means includes user-defined exception support generating means for generating hardware supporting that user-defined exception as part of the processor hardware implementation.

49. (Original) A processor simulation system comprising:

hardware simulation means for executing a hardware description of an extensible processor;

software simulation means for executing a software reference model of the extensible processor; and

cosimulation means for operating the hardware simulation means and the software simulation means and comparing results of simulations therefrom to establish correspondence between the hardware description of the extensible processor and the software reference model of the extensible processor.

50. (Previously Presented) The processor simulation system of claim 49, wherein the hardware description of the extensible processor includes a description of hardware supporting one or more user-defined instructions.

51. (Previously Presented) The processor simulation system of claim 50, wherein the software reference model includes a model of the extensible processor including the one or more user-defined instructions.

52. (Previously Presented) The processor simulation system of claim 49, wherein the hardware description of the extensible processor executed by the hardware simulation means includes a description of one or more user-defined states.

53. (Previously Presented) The processor simulation system of claim 52, wherein the software reference model executed by the software simulation means includes a model of the extensible processor including the one or more user-defined states.

54. (Previously Presented) The processor simulation system of claim 53, wherein the cosimulation means establishes correspondence of all the processor states, including the user-defined states, between the hardware description of the extensible processor and the software reference model of the extensible processor..

55. (Previously Presented) The processor simulation system of claim 49, wherein the hardware description of the extensible processor executed by the hardware simulation means includes a description of one or more user-defined register files.

56. (Previously Presented) The processor simulation system of claim 54, wherein the software reference model executed by the software simulation means includes a model of the extensible processor including the one or more user-defined register files.

57. (Previously Presented) The processor simulation system of claim 56, wherein the cosimulation means establishes correspondence of all the processor register files, including the

user-defined register files, between the hardware description of the extensible processor and the software reference model of the extensible processor.

58. (Previously Presented) The system of claim 39, wherein the user-defined portion of the configuration specification supports one or more user-defined instructions.

59. (Previously Presented) The system of claim 39, wherein the instructions used in the software development tools include one or more user-defined instructions.

60. (Previously Presented) The system of claim 39, wherein the pipeline logic, pipeline stalling logic and instruction rescheduling logic are for supporting one or more user-defined instructions.

61. (Previously Presented) The system of claim 45, wherein the user-defined portion of the configuration specification supports one or more user-defined instructions.

62. (Previously Presented) The system of claim 45, wherein the processor instruction set includes a predetermined portion and a user-defined portion.

63. (Previously Presented) The system of claim 48, wherein the user-defined portion of the configuration specification supports one or more user-defined instructions.

64. (Previously Presented) The system of claim 48, wherein the processor instruction is a user-defined instruction.